The Universal Data Transcriber. A New Approach to Data Conversion Equipment

MARVIN S. MAXWELL
NONMEMBER AIEE

The Universal Data Transcriber is a device to convert digital data from one form to another. The input to the system can be cards, paper tape, magnetic tape, or the output of almost any source of digital data that does not exceed 500,000 bits per second. The output of the converter can be cards, magnetic tape, paper tape, low- or high-speed printers, point plotters, or almost any other digital output device. The conversion process allows for very flexible code and format conversion as the data is transcribed from one medium to another.

The Universal Data Transcriber consists of appropriate input-output equipment and a microprogrammed, semi-variable-structure, stored-programmed, single-address binary computer. The computer has a high-speed random access memory of 8,192 characters of 8 data bits plus one check bit each, and with average 70,000 instructions per second. The operation of the system depends upon the microprogram of the computer to generate special orders which will transfer data from the particular external input device currently in use to the computer memory, and from the memory to the external output device currently in use. The use of microprogramming, which is accomplished by use of a plugboard, allows an efficient transfer of data between the computer memory and the external devices with a minimum of special equipment. Conversion of the data within the memory from one form to another is accomplished by the use of an appropriate stored program. This gives a very flexible system, since all that is required to change the system from one job to another is to change the connections to the external equipment, insert a different plugboard, and load a new program into the computer memory. This system was conceived, designed, and is under construction by the Computer Research and Development Branch of the Computation and Exterior Ballistics Laboratory of the United States Naval Proving Ground, Dahlgren, Virginia.

Description of the Computer

The basic structure of the computer is shown in the diagram of Fig. 1. It con-
sists of an input register, output register, two computing registers (R-1 and R-2), six B-registers (address modifiers), instruction register, instruction counter, indicator latches (single-bit registers), and other special registers. External devices communicate with the computer via the input and output registers under control of the computer. The input register can select at high speed from either of two different external devices. The output register is normally connected to only one unit. Indicator latches are used both to control the external devices and to signal the condition of the external devices to the computer. Special electronic signal-generating equipment tailored to each type of external device is used to facilitate communication with the input register, output register, indicator latches, and the external device.

The computer has a high-speed random access core memory with a capacity of 8,192 characters, each consisting of eight bits plus a check bit. A character is treated as a pure binary number by the arithmetic section of the computer and a single instruction generally affects only one character. Since there are no multiply or divide orders, the operating binary point may be considered to be in any convenient location. The carry (borrow) bit may be propagated from character to character in addition (subtraction) with use of double precision orders. A single reference to the memory brings out four characters designated as M0, M1, M2, and M3 into the memory register. Addresses evenly divisible by four always correspond to the character read out as M0. Instruction words consist of the four characters M0, M1, M2, and M3. Instruction words are logically divided into 4 fields as shown, namely: operation code, B-register specification, address specification of reference to memory, and the limit value of Bx.

The time required to execute orders which require only one reference to memory such as program transfers is approximately 11 microseconds. Orders which require two references to memory, such as read out or store orders, require approximately 21 microseconds. A complete description of the order codes is available in Appendix I.

Each of the six B-registers has eight binary columns plus a check column. Address modification is performed by overlaying the contents of the eight least significant columns of the 13 binary-column address field with the contents of the specified B-register. The overlaying process is a logical addition in which the contents of a binary column in the address field is changed only when a zero is overlaid with a one-bit.

For example:

<table>
<thead>
<tr>
<th>Address field</th>
<th>1</th>
<th>1011</th>
<th>1001</th>
<th>0110</th>
</tr>
</thead>
<tbody>
<tr>
<td>B-register</td>
<td>1</td>
<td>0101</td>
<td>0101</td>
<td>0111</td>
</tr>
</tbody>
</table>

The contents of B-registers may be incremented by one either automatically after use if so desired or on command. If it is desired to increase the contents of a B-register by any value other than one or to decrease its contents by any value, the B-register may be operated on by the use of appropriate order codes.

One of the B-registers, B0, is normally connected as a real-time counter. It may be preset to any value up to its limit of eight bits and a one is added to the contents of the register every 0.504 millisecond. The counting ceases when the contents of the register reaches 377 octal. With use of this feature and instructions for read out and limit sensing on the B-registers it is possible to program measurement of real-time for controlling external devices.

### Special Features of the UDT

The most outstanding difference between the computer of the Universal Data Transcriber (UDT) and any other single-address binary computer is the availability of the plugboard and the plugboard instructions. The plugboard is divided into three regions. The first region consists of information coming from equipment in the computer to the plugboard. This includes all of the registers, such as register 1, register 2, input register, output register, instruction register, instruction counter, B7, and the indicator latches, plugboard instruction specification, and the internal clock. Also in this region are external inputs from the various input and output devices which have been converted to the proper signal levels. The second region of the plugboard consists of a set of approximately 75 logical packages. These packages are identical to those used in the construction of the rest of the computer. In the third region of the plugboard are exits from the plugboard to the control lines in the computer. These lines control the transfer of data from "register to register," use of the B-registers, controlling memory cycles, setting of indicator latches, shifting various registers, etc.

Mann—The Universal Data Transcriber
three regions of the plugboard almost any conceivable, or desirable, cycle of actions can be controlled from the plugboard. This feature is primarily for use with external devices to get data to or from them and the memory of the UDT.

The indicator latches in the computer are used primarily for communication between the UDT and external devices. For example, some of the indicator latches could be wired, via the plugboard, to control the stopping, starting, or reading or writing of a tape unit. Other indicator latches could be used to indicate to the UDT that an external device is in certain conditions; for example, that a card reader is moving cards, or ready to scan one row of information, or that it is out of cards, etc. Thus, the program can control external devices, and external devices can be sensed by the program by use of the indicator latches.

Another feature of the UDT is the "program interrupt" ability. If a particular exit on the plugboard is energized, the computer will go into a program interrupt cycle. This exit can be energized from an indicator latch, or combinations of indicator latches and various conditions by wiring on the plugboard. When this condition occurs the computer will automatically make a program transfer to instruction location 4 at the end of the current instruction. The address, Y, of the instruction which would have normally been executed next, if the program interrupt condition had not occurred, will be automatically stored in character locations 1 and 2 in a form so that if the character in location 0 is the code for a program transfer, jump, command and the instruction at location 0 were to be executed, the computer would jump to the proper address, Y. When this feature is used the program, starting at location 4, must be suitable to take the appropriate action for the condition which caused the jump. After this is done, the program would normally remake the appropriate registers, and then jump to location 0, which would cause the jump back to the main program at the proper place. By using this feature the computer can react rapidly to external control information without requiring repeated sensing on the condition.

Advantages and Limitations

The major advantage of the UDT is its flexibility. It is not tailored to any specific computer or type of data conversion and is therefore not likely to become obsolete as fast as many specialized converters. The microprogramming and stored program features make it easy to implement almost any desired conversion with a minimum of engineering effort and special equipment. The major disadvantage of this approach is that it is more expensive than any single specialized converter.

To establish the capabilities of the UDT, several preliminary programs have been prepared. One program for converting 80-column alphanumeric International Business Machines Corporation (IBM) cards to NORC magnetic tape

Maxwell—The Universal Data Transcriber
provides for arbitrary code and format conversion, specified by header cards, and converts data to magnetic tape at a rate of 450 cards per minute. Similar programs have been developed for conversion from one magnetic tape system to another. If there is a conversion in both the code representation of the data and in the format, but not in the number base, the system can convert 4-, 5-, 6-, 7-, or 8-bit characters from one form to another at a rate of approximately 3,000 characters per second. Conversion can be made from 48-bit binary words to decimal digit words at a rate of approximately 16 words per second. Conversion can be made from 13-digit decimal words to binary words at rates in excess of 50 words per second.

Engineering Considerations

The Universal Data Transcriber is being designed and constructed at the United States Naval Proving Ground, Dahlgren, Virginia. Subcontractors are providing the memory, logical building blocks, and various specialized input and output circuitry. The system which is currently being assembled is scheduled to become operational in the first quarter of 1959 and will consist of the central computer, a magnetic tape transport, an IBM 514 used as a printer, a modified IBM 101 used as a card reader, and a Flexowriter. The computer with memory, level conversion circuitry, plugboard, power supplies, and console will occupy 5 relay racks.

The logical building blocks are all transistorized megacycle SEAC type circuitry built by Computer Control Company. Some of these are being modified to provide 2-phase operation where the extra speed is required. The memory is an all-transistorized magnetic core memory with a full read-write cycle time of 10 microseconds, and operates in parallel on a 36-bit word or 4 characters of 9 bits each. The 80-brush reading station of the IBM 101, used as a 450-card per minute reader, will load the data from a row in the card in parallel into a magnetic shift register which will be shifted into the computer on four wires in 600 microseconds. A similar circuit will be used on the second reading station so as to provide a check on the reading. Data are punched into IBM cards at 100 cards per minute by serially shifting, one bit at a time, at a 100,000 cycle shift rate, the 80 bits in the row to be punched. This shift register will pick up relays which will control the punch magnets in an IBM 514. The reading station which follows the punching station will be equipped with magnetic shift register for reading back the data from the punched card for a check. The same shift register and relays which are used in punching is 120-bits long so that it can be used to control the printing on an IBM 407. A Flexowriter is permanently attached to the system to provide communication between the computer and the operator, and is used as an input for the program tapes, and as an output or input of 5-, 6-, 7-, or 8-channel paper tape.

Conclusions

The Universal Data Transcriber is a very flexible data conversion system. This system not only allows conversion of data from almost any digital source into an appropriate form for a general-purpose computer but also reduces the computing time required in the general-purpose computer, since most of the format conversion and editing will be performed in the Universal Data Transcriber.

Appendix I. Order Codes

An instruction word consists of 32 bits. 13 bits are used to designate the address of the operand; this is called the address field and is designated by A'. Three bits specify which of the six B-registers, B1, B2, B3, B4, B5, or B6, (also called B-box, index register, or address modifier) is to be used. In all cases the contents of the selected B-register (8 bits) is overlaid, forming the logical sum, with the low-order 8 bits of the A field before the address is used. Eight bits are used to designate the limit value of the contents of the selected B-register to control some conditional transfers. Eight bits, divided into 3 digits, are the command field. The high-order digit specifies the code group 0, 1, 2, or 3. An instruction word is assembled from 4 characters as indicated.

The prime symbol "'" is used to designate "the contents of." For example, a reference to A' refers to the 13-bit address field itself, but a reference to A"' refers to the character stored at address A. In all cases, unless specifically stated to the contrary, when data is transferred from one register to another the data remain in the source register and the receiving register is cleared to all zeros before, or as, the data are received.

In writing the orders in the following groups the subject matter is consistent with the concepts expressed in the following definitions:

(a) The 8-bit character made available by access to the memory at address A. "A" is the 8-bit character made available by access to that B-box which is designated as Bx.

The address A may be modified by overlaying it with Bx'. A' will then identify the 8-bit character made available by access to the memory at the modified address A.

<table>
<thead>
<tr>
<th>Bits</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>3-digit specification of command field</td>
</tr>
<tr>
<td>17</td>
<td>5-digit specification of limit value of Bx</td>
</tr>
<tr>
<td>16</td>
<td>1-digit Bx specification</td>
</tr>
<tr>
<td>15</td>
<td>3-digit specification of address field</td>
</tr>
<tr>
<td>14</td>
<td>2-digit command field</td>
</tr>
<tr>
<td>13</td>
<td>13-bit address in the instruction</td>
</tr>
<tr>
<td>12</td>
<td>A' = The 8-bit character made available by access to the memory at address A.</td>
</tr>
<tr>
<td>11</td>
<td>Bx' = The 8-bit character made available by access to that B-box which is designated as Bx.</td>
</tr>
</tbody>
</table>

Group 0

The execution time of orders in this group is 20 microseconds. All A' addresses may be overlaid with the contents of any one of six B registers (in the recirculating ring) by specifying Bx in the B-field of the instruction. The value of x may be: 1, 2, 3, 4, 5, or 6 to overlay A with the contents of B1, B2, B3, B4, B5, or B6 respectively.

If B0 is specified, there will be no overlay. If B7 is specified, the contents of Register B7 will not be overlaid on A. Subtract orders do not leave a record of a possible negative sign in the result. Propagate and subtract orders are used in multiple precision arithmetic.

041 A' to R1: clear register 1, transfer A' into register 1.

051 A' to R1: add 1 to Bx after modifying A.

021 A' to R2: clear register 2, transfer A' into register 2.

031 A' to R2: add 1 to Bx after modifying A.

025 Add A' to R1: clear register 2, transfer A' into register 2, clear carry storage, add A' to contents of register 1, put sum in register 1, store overflow bit, if any, in carry storage, reset register 2 to all zeros.

Maxwell—The Universal Data Transcriber
035 Add \( A' \) to \( R1 \); add 1 to \( Bx' \) after modifying \( A \).

036 Logical product (extract) \( R1' \); clear register 2, transfer \( A' \) into register 2, add \( A' \) to contents of register 1 without previously resetting borrow storage, put sum in register 1, store overflow bit, if any, in carry storage, transfer register 2 to all zeros.

037 Add \( A' \) to \( R1' \) with no internal carry; add 1 to \( Bx' \) after modifying \( A \).

038 Overlay \( R1' \) with \( A' \) and add the result in \( R2' \); add 1 to \( Bx' \) after modifying \( A \).

039 Store \( R1' \) in \( A \); add 1 to \( Bx' \) after modifying \( A \).

Note: With reference to orders 023, 023, 022 and 032, if the contents of register 1 are greater than or equal to the contents of register 2 (minuend greater than or equal to subtrahend) the result will be a binary difference with final contents of borrow storage cleared to zero. If, however, the contents of register 2 are greater than the contents of register 1 (minuend less than subtrahend) there will finally be a one in borrow storage and the true difference will be equal to the contents of register 1 minus 2\(^n\). In this latter case, by subtracting the final contents of register 1 from zero, the absolute value of the difference between the minuend and the subtrahend can be obtained.

For multiple-precision arithmetic the nonreset of the borrow in orders 022 and 032 is related to preceding orders so that the final presence or absence of borrow refers to the entire sequence of operations. The true difference may be equal to the contents of register 1 minus 2\(^n\) or 2\(^n\), etc.

033 Subtract \( A' \) from \( R1' \); clear register 2, transfer \( A' \) into register 2, clear borrow storage, subtract \( A' \) from contents of register 1, store the borrow bit, if any, in borrow storage, store the answer in register 1, reset register 2 to all zeros. (See note.)

034 Subtract \( A' \) from \( R1' \); add 1 to \( Bx' \) after modifying \( A \). (See note.)

035 Propagate subtract \( A’ \) from \( R1' \); clear register 2, transfer \( A' \) into register 2, subtract \( A' \) from contents of register 1 without resetting borrow storage, store the subtrahend bit, if any, in borrow storage, store the result in register 1, reset register 2 to all zeros. (See note.)

036 Logical product (extract) \( A' \) and \( R1' \); clear register 2, transfer \( A' \) into register 2, form the logical product, bit by bit, of \( A' \) and contents of register 1, store the result in register 1, clear register 2.

Example:
- Contents of register 1 = \( R1' = \) 10101010
- Contents of register 2 = \( A' = \) 00001111

Logical product = 00001010

037 Logical product (extract) \( A' \) and \( R1' \); add 1 to \( Bx' \) after modifying \( A \).

038 Add \( A' \) to \( R1' \) with no internal carry: reset carry storage, clear register 2, transfer \( A' \) into register 2, add \( A' \) to contents of register 1 with no internal carry propagation during adding, store the result in register 1, store the high-order end of register 2. The presence of a final result in register 1 is an indication that the original operands were not equal.

Example:
- Original \( R1' = \) 01011101
- \( R2' = A' = \) 10111010
- Final \( R1' = \) 11001000

Maxwell—The Universal Data Transciber
A Universal Computer Language Translator

R. B. Bonney
Nonmember AIEE

Can Univac talk to Bizmac? As more and more electronic data-processing systems are put into operation, the need for rapid and efficient information interchange becomes important. Many organizations have evolved data-processing systems using a combination of different computer types, together with a variety of input and output equipment. The use of different computers in a single organization does not necessarily represent poor planning, since the requirements of one group may dictate the use of a particular type of computer which is inadequate for the requirements of a different group.

To achieve efficient utilization of an electronic data-processing system, it is often desirable to transfer output data from one computer to a different computer for additional operations. The need for interchange between different organizations is also beginning to develop, and can be expected to increase as electronic data-processing spreads. Acceptance of payroll deduction reports on International Business Machines Corporation (IBM) 705 magnetic tapes by the Treasury Department and the Social Security Agency are typical examples.

Transformation from one form of data to another is another common requirement. Conversion from magnetic tape to punched paper Teletype tape is representative of a wide variety of such requirements.

Unfortunately, the designers of electronic computers have made very little attempt at standardization. In nearly all cases, the recording format and coding used by the various computer manufacturers are sufficiently different to prevent direct interchange of data. Undoubtedly, the very rapid development and growth of the computer industry have been major factors in the lack of standardization.

The commonly used method of accomplishing data translation between computers up to now has been by means of punched cards, and in some cases punched paper tape. Translation by these methods is seriously hampered by the speeds of operation. Some special-purpose high-speed translation equipment has been built for direct translation of data on magnetic tape. However, because of the wide variation in requirements, the equipment developed for a specific application ordinarily cannot be used in a different application without modification or complete redesign.

Realization of the need for "universal" translation capability resulted from work on data-handling systems for scientific test data. For this purpose, a computer format control buffer was developed to accept unsynchronized, continuous digital input data which are to be converted to magnetic tape, in a format directly usable by a computer.

In discussing other applications of this equipment, it soon became evident that no two applications were alike, and that too much re-engineering would be required to adapt the equipment to each new situation. It was, therefore, decided that an attempt should be made to develop a system with the flexibility necessary to meet all commonly encountered translation requirements without becoming overcomplex and economically impractical. The Computer Language Translator is the result.

In considering all possible applications, there is a wide variety of requirements.

R. B. Bonney is with Electronic Engineering Company of California, Santa Ana, Calif.

Bonney—A Universal Computer Language Translator